

# UML

## El Lenguaje de Modelado Unificado



---

Maestría en Ingeniería de Software



# Agenda

- Model Driven Architecture (MDA)
- **Unified Model Language (UML)**
- Object Constraint Language (OCL)
- Patrones
- Conclusiones

# Contenido

## **Unified Modeling Language (UML)**

- Introducción a UML
  - Definición
  - Origen y evolución
  - Bloques de construcción
- Diagramas UML
  - Diagramas estructurales
  - Diagramas de comportamiento
  - Diagramas de interacción

# Contenido

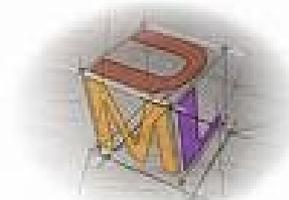
## **Unified Modeling Language (UML)**

- Diagramas de clases
  - Clases
  - Relaciones
- Metamodelado
  - Jerarquía de modelos
  - MOF
  - Metamodelo UML
- Perfiles UML
  - Definición
  - Aplicación
  - Perfiles en MDA

# Introducción a UML

- El diseño y modelado de sistemas ocupan un importante lugar entre los ingenieros de software.
- Para la descripción de un modelo, es necesario tener un lenguaje con una sintaxis y semántica precisa.

**UML permite fácilmente especificar, construir,  
visualizar y documentar artefactos de sistemas de  
software**



# Introducción a UML

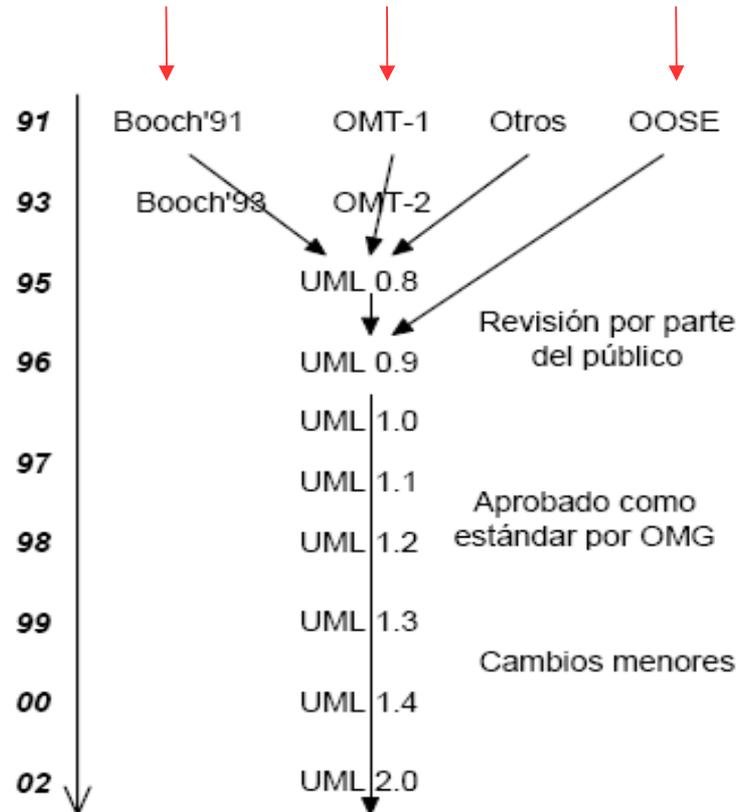
## Objetivos de UML

- Unificar lenguajes y técnicas OO existentes
- Incorporar buenas prácticas de la industria
- Fácil de aprender y rico visualmente
- Que sea de propósito general

# Introducción a UML

## Origen y evolución

Grady Booch Jim Rumbaugh Ivar Jacobson



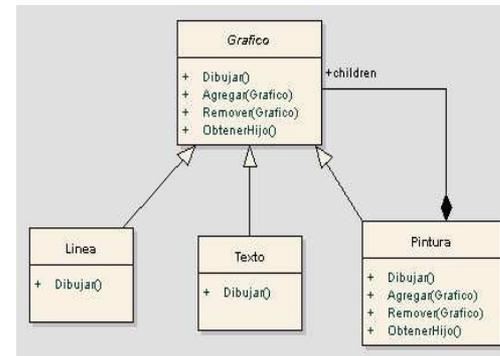
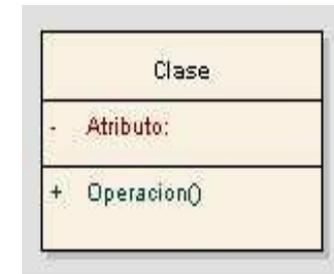
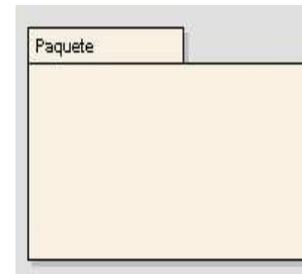
# UML

## Bloques de construcción

**Elementos** (estructurales, de comportamiento, de agrupamiento y anotacionales)

**Relaciones** (dependencia, asociación, generalización, realización y agregación)

**Diagramas** (casos de uso, de clase, de comportamiento, de estado, de actividad, de interacción, de implementación, de componentes, de despliegue)



# UML

## Diagramas

UML 2.0 define 13 tipos de diagramas:

### Diagramas Estructurales:

- diagrama de clases,
- diagrama de objetos,
- diagrama de componentes,
- diagrama de estructura compuesta,
- diagrama de paquetes,
- diagrama de despliegue

### Diagramas de comportamiento:

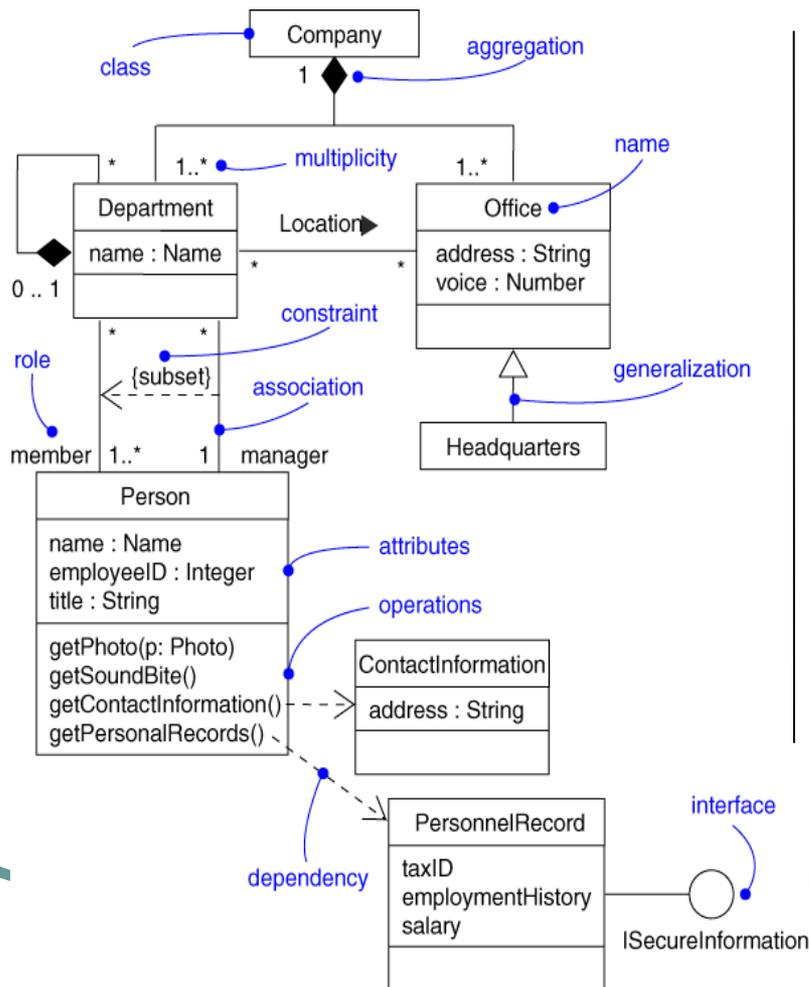
- diagrama de casos de uso,
- diagrama de actividades
- diagrama de máquina de estados

### Diagramas de interacción:

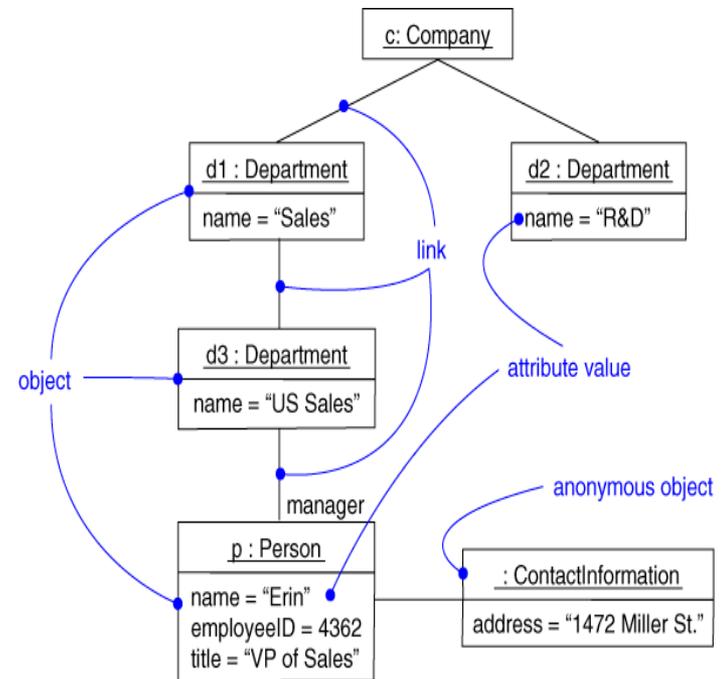
- diagrama de colaboración,
- diagrama de secuencia,
- diagrama de tiempo,
- diagrama general de interacciones

# Diagramas Estructurales

## Diagrama de clases

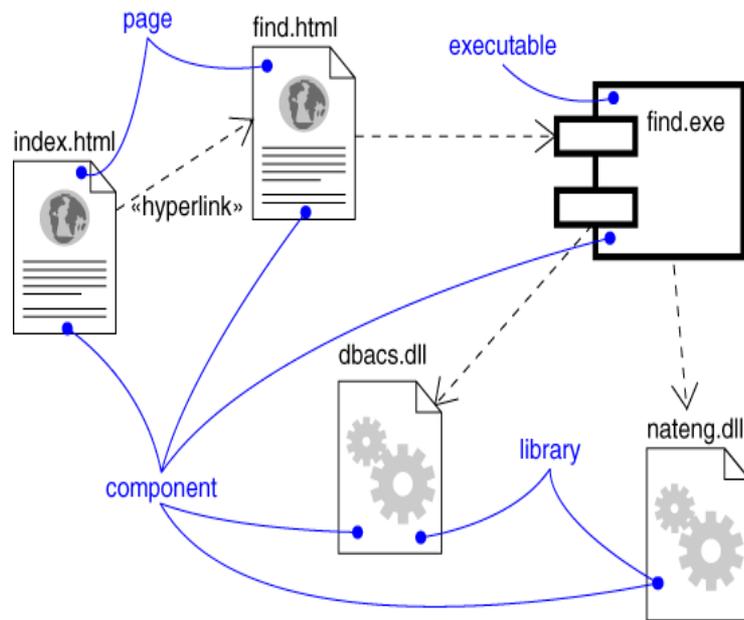


## Diagrama de objetos

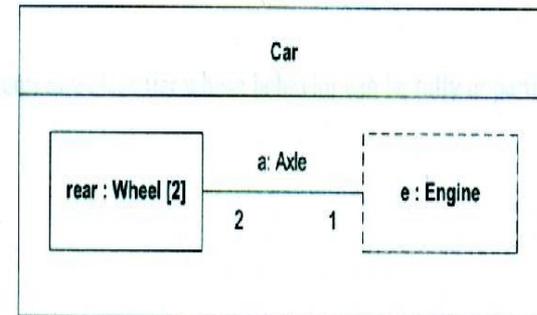


# Diagramas Estructurales

## Diagrama de componentes

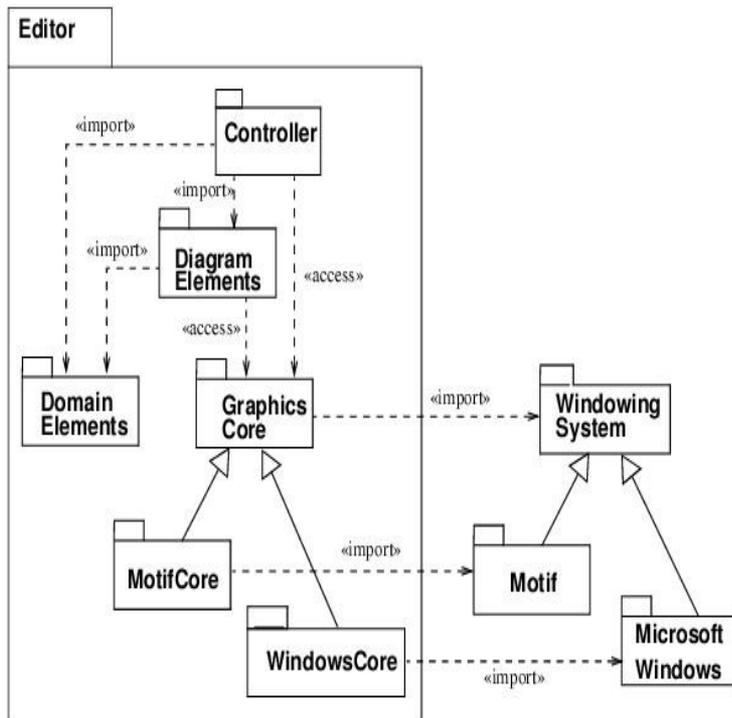


## Diag. estructura compuesta

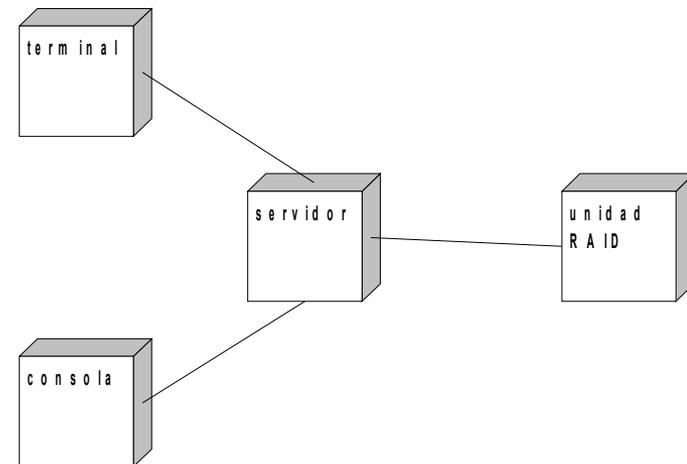


# Diagramas Estructurales

## Diagrama de paquetes

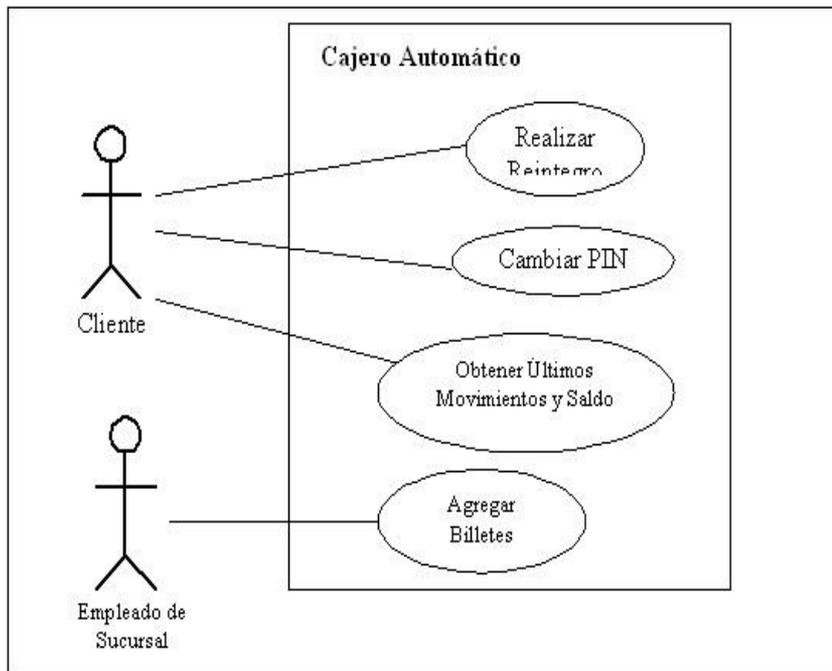


## Diag. de despliegue

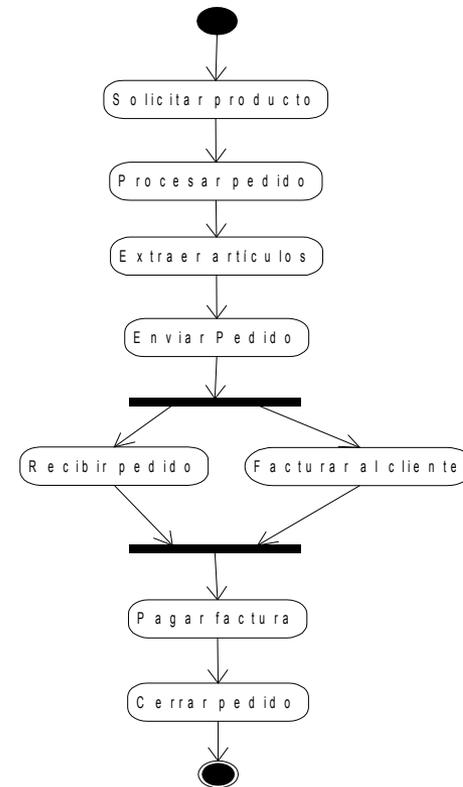


# Diagramas de Comportamiento

## Diagrama de casos de uso

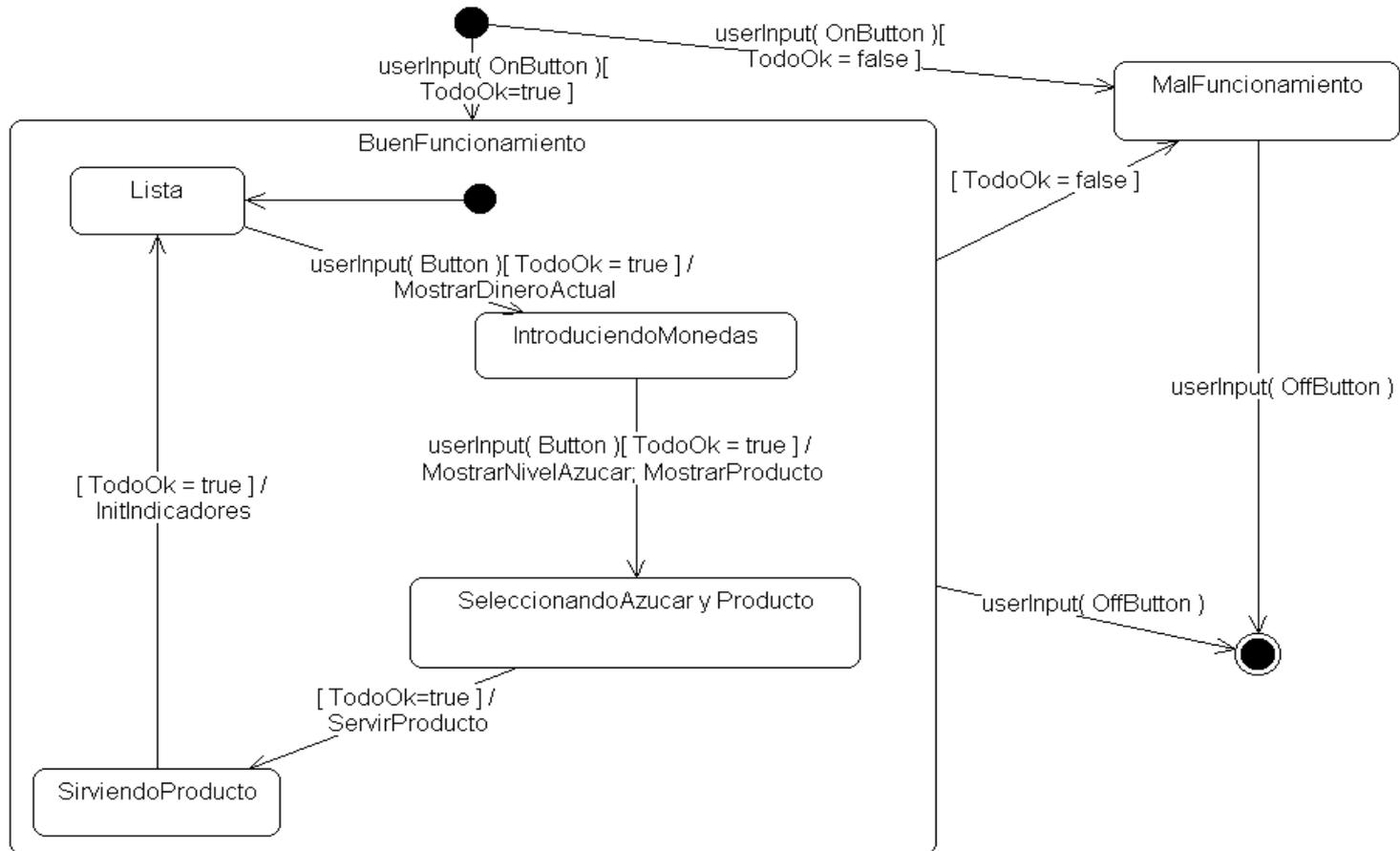


## Diag. de actividades



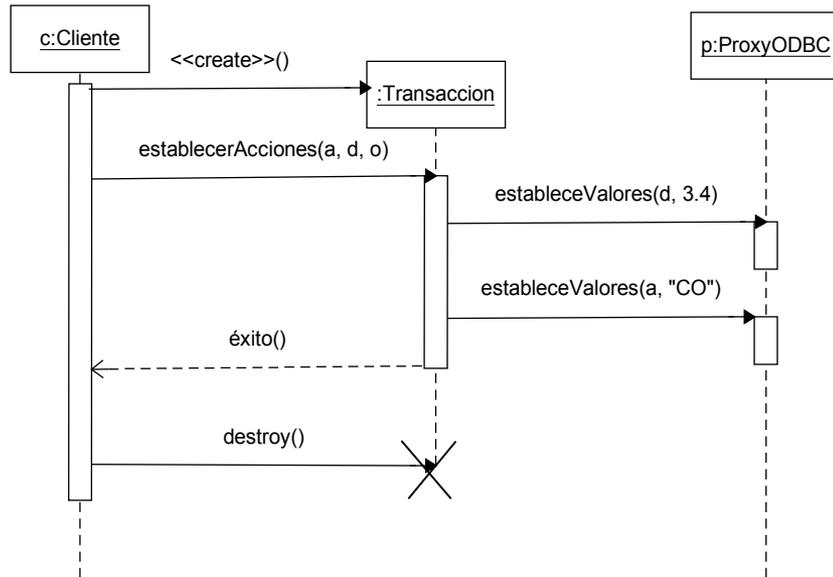
# Diagramas de Comportamiento

## Diagrama maquina de estados

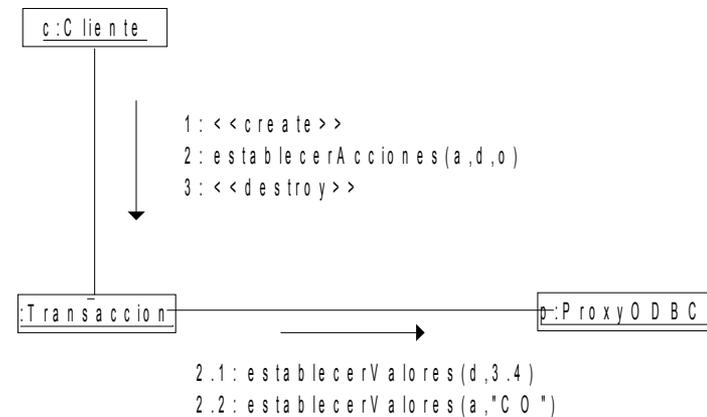


# Diagramas de Interacción

## Diagrama de secuencia

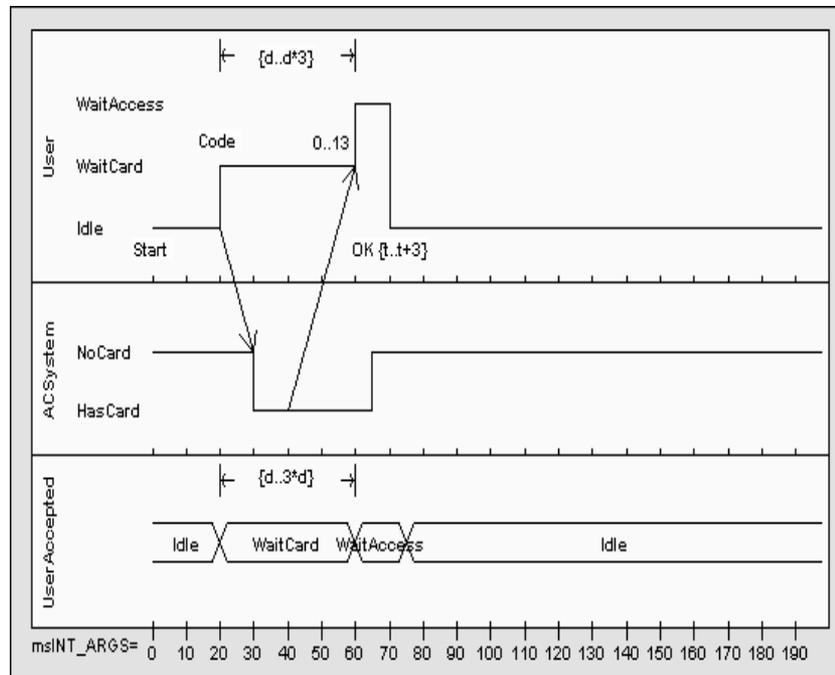


## Diagrama de colaboración

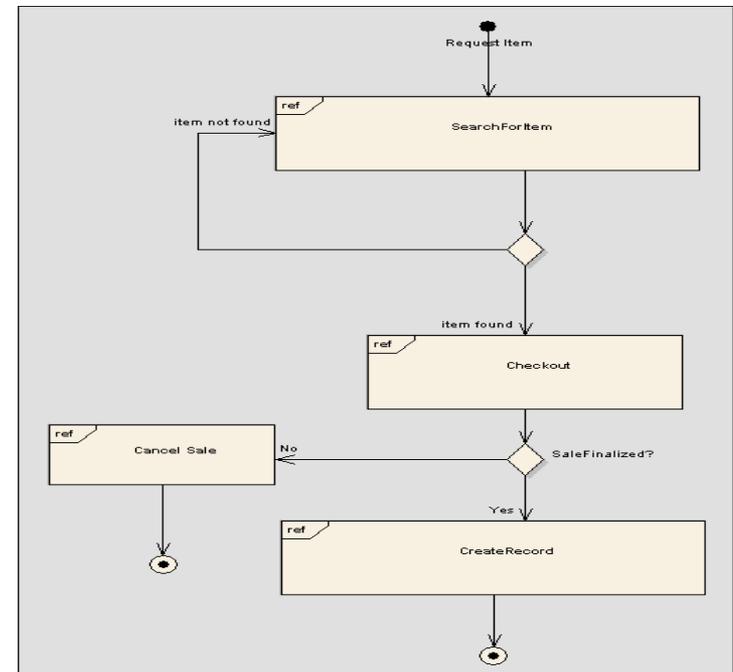


# Diagramas de Interacción

## Diagrama de tiempo



## Diagrama grl. de interacciones



## Unified Modeling Language (UML)

- Introducción a UML
- Diagramas UML
- **Diagramas de clases**
- Metamodelado
- Perfiles UML
- Perfiles en MDA

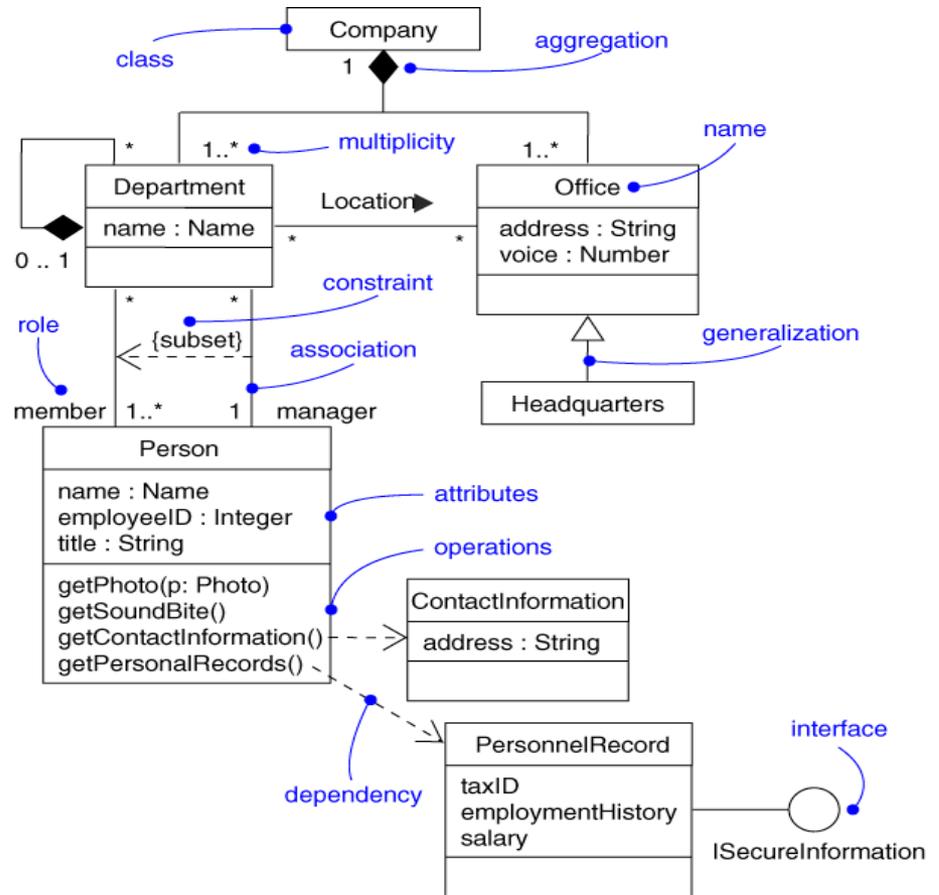
# Diagramas de clases

## Clases

- nombre
- atributos
- métodos

## Relaciones

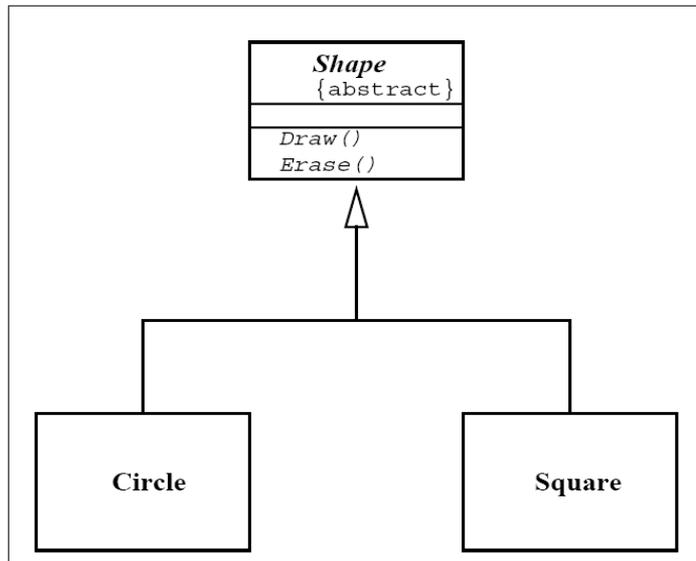
- nombre
- rol
- multiplicidad
- navegabilidad



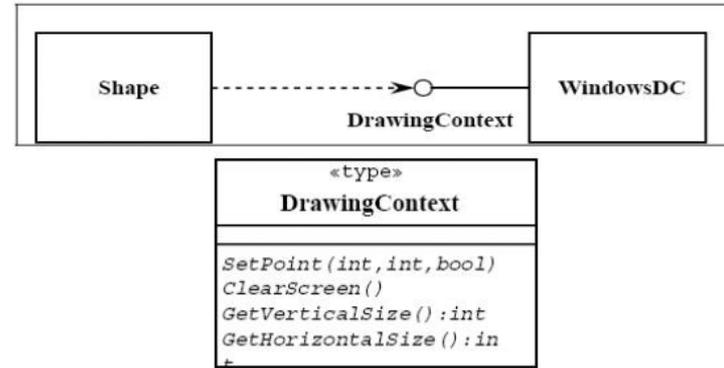
# Diagramas de clases

## Clases

### Clase abstracta



### Interface

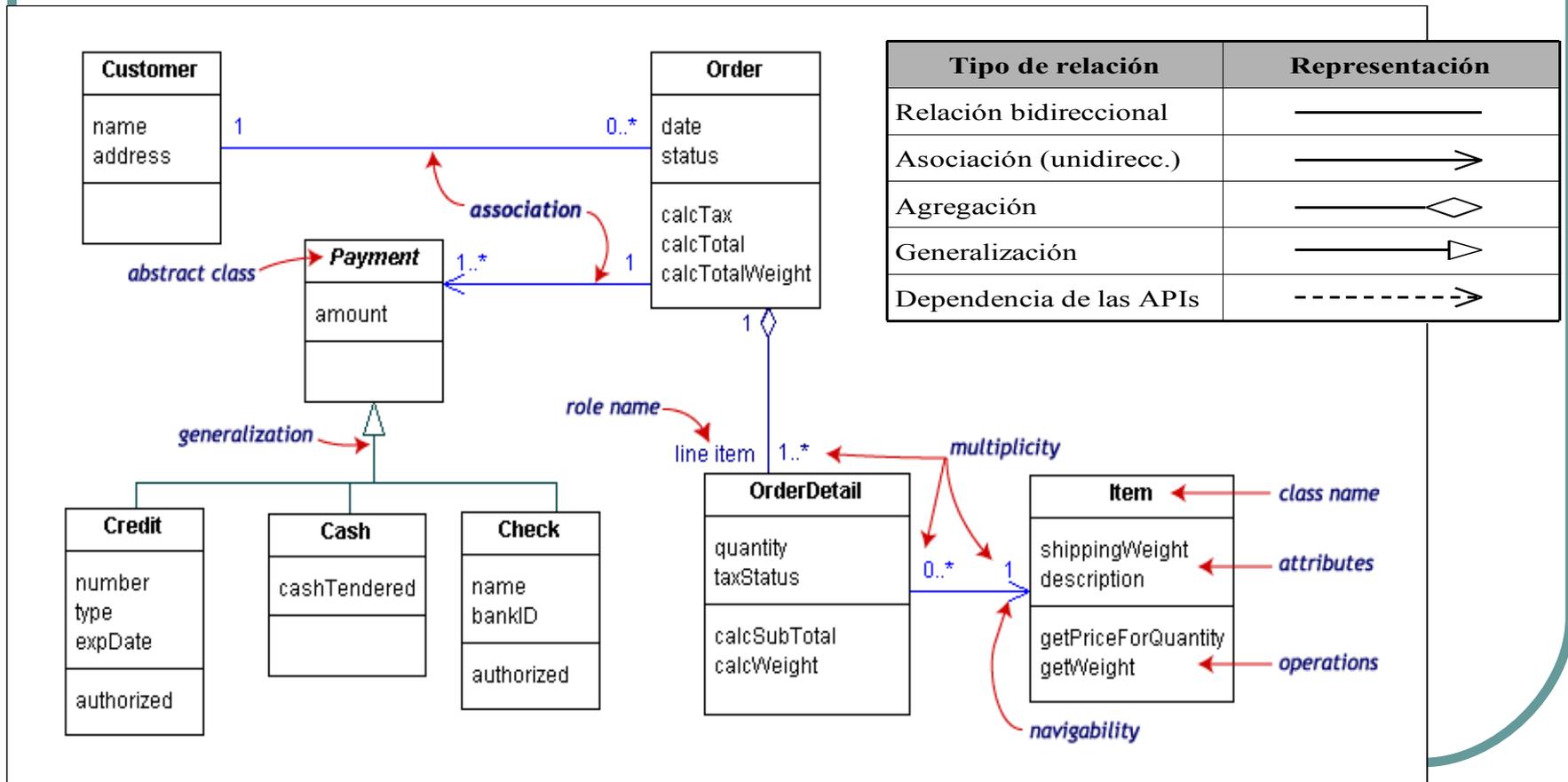


### Clase parametrizada



# Diagramas de clases

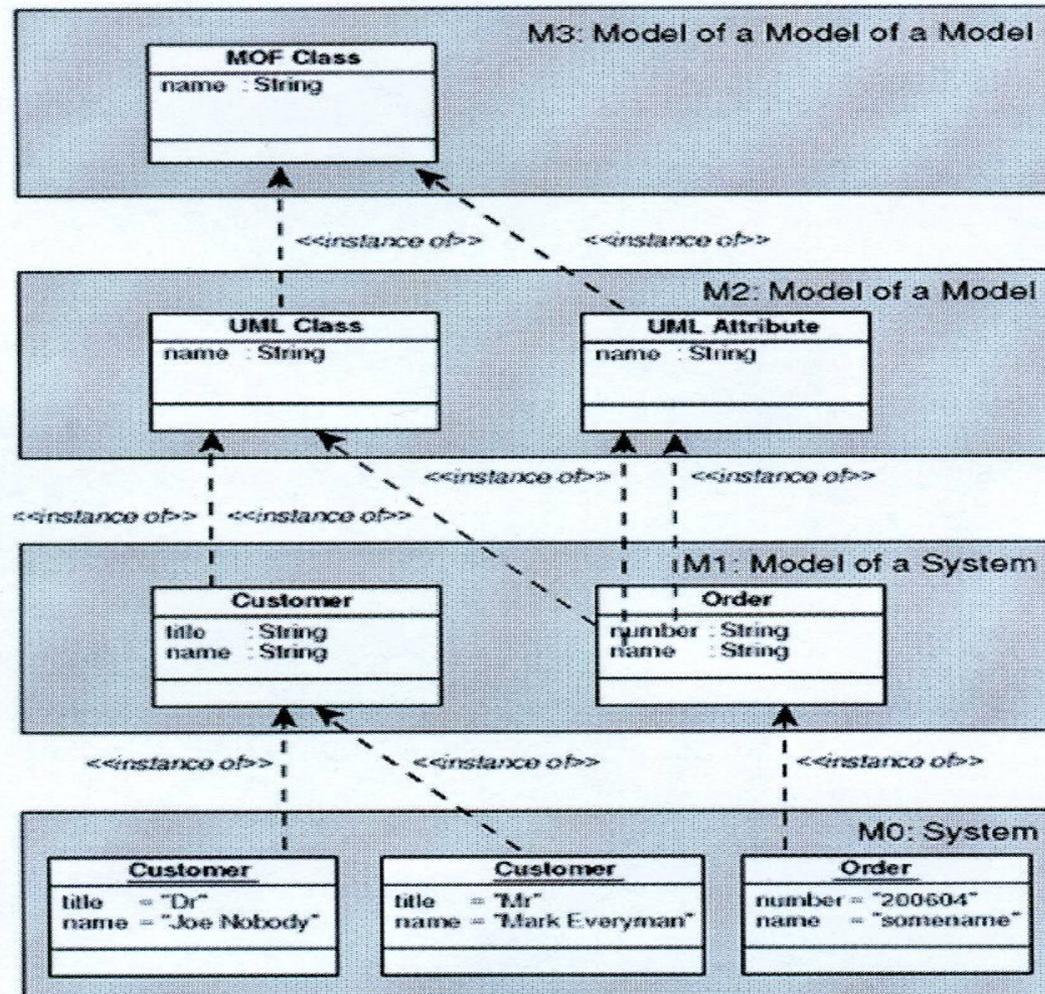
## Relaciones



## Unified Modeling Language (UML)

- Introducción a UML
- Diagramas UML
- Diagramas de clases
- **Metamodelado**
- Perfiles UML
- Perfiles en MDA

# Metamodelado



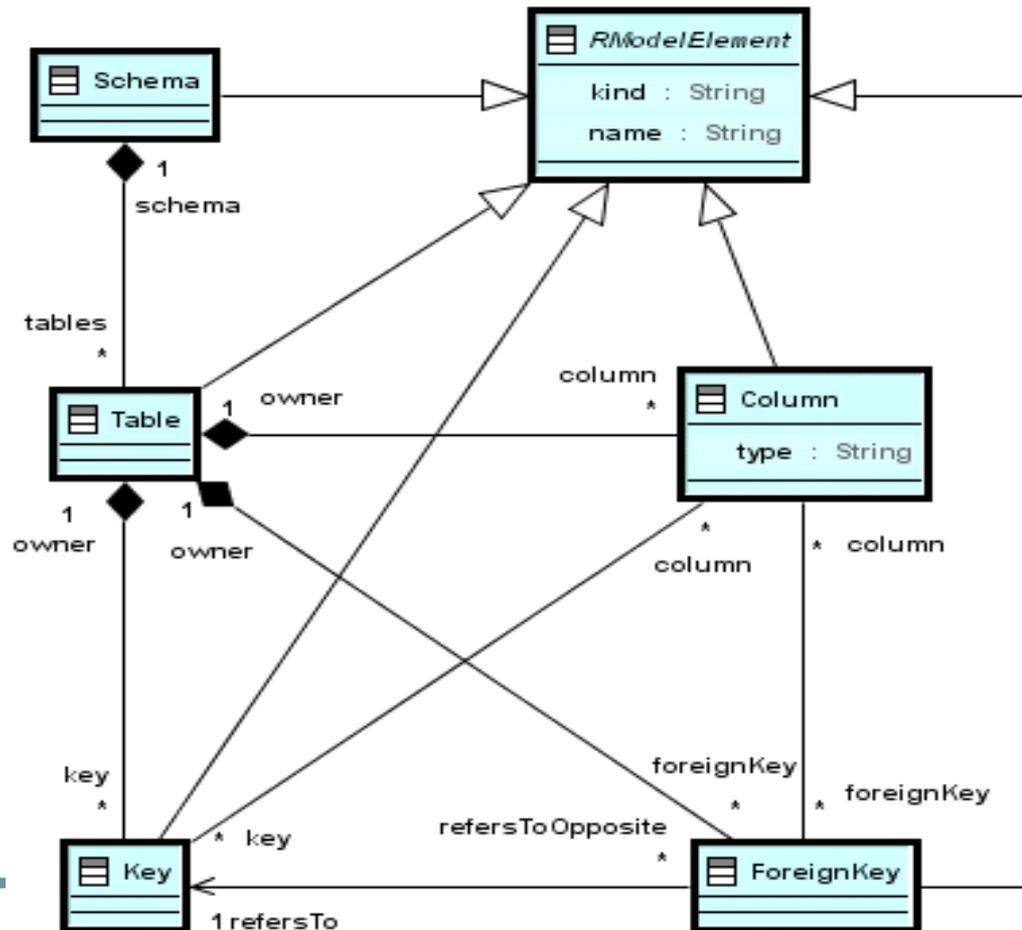
# Metamodelado

## MOF

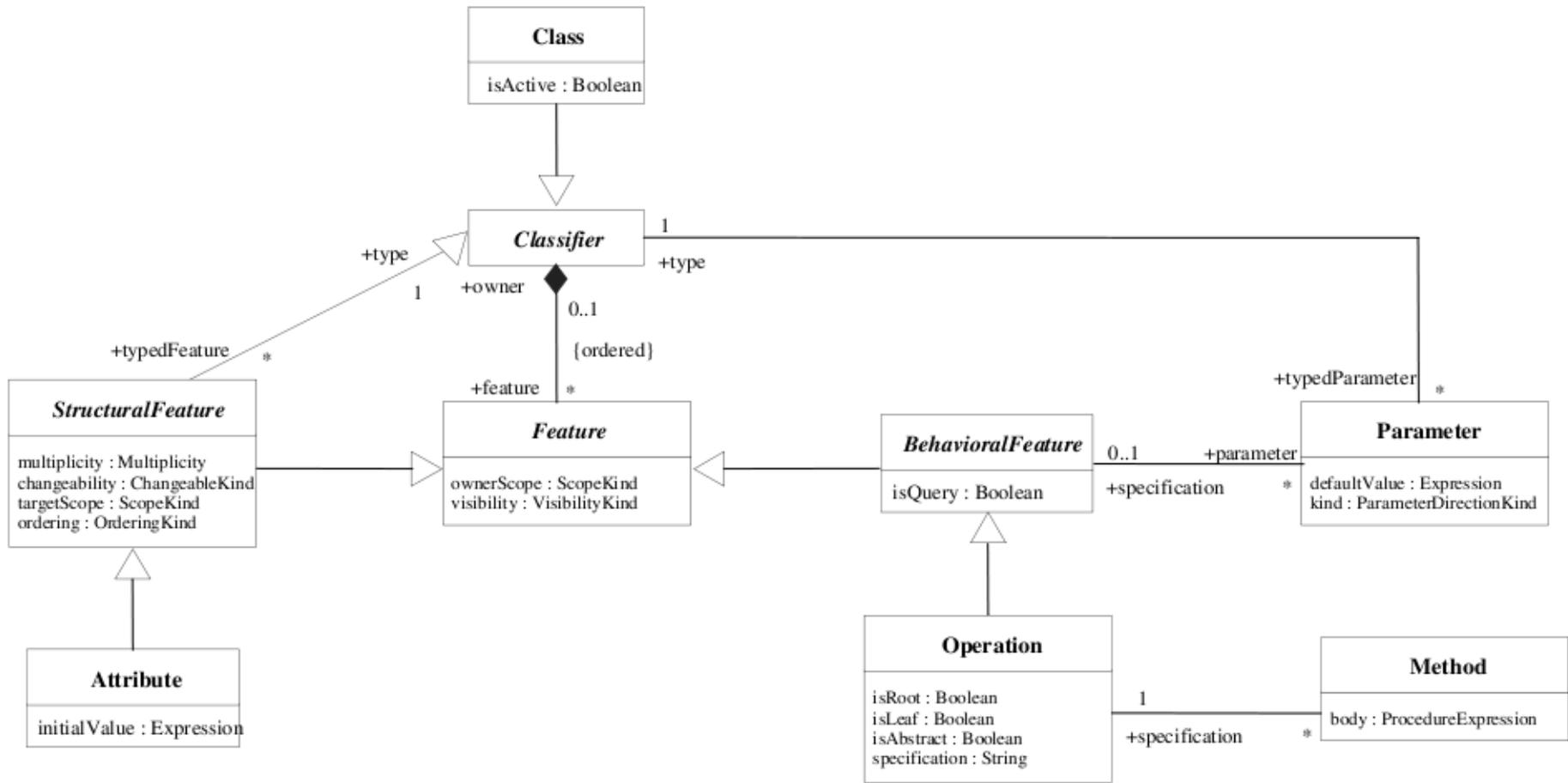
- MOF es un metamodelo que se define a sí mismo
- Lenguaje estándar de la OMG para definir la estructura de metamodelos
- Definir estándar para mapeo entre metamodelos
- No solo el metamodelo UML puede ser definido con MOF

# Metamodelado

Definición de un metamodelo para RDBMS



# Metamodelo UML

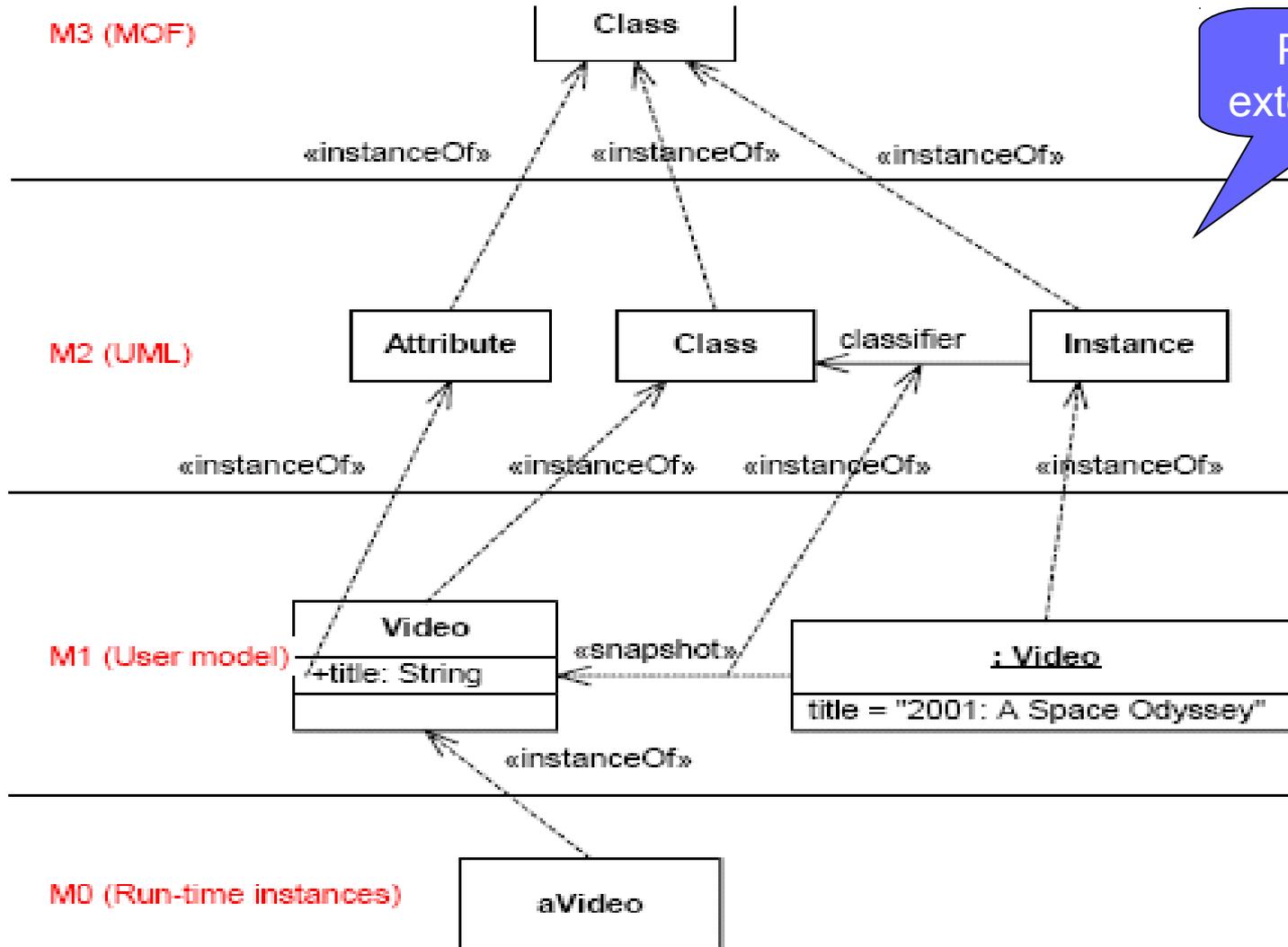


# Metamodelo UML

- Posibilidades para definir un lenguaje:
  - Crear un metamodelo con MOF
  - Crear un perfil UML

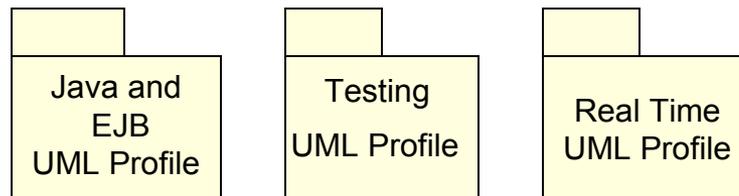
# Perfiles UML

## Jerarquía de metamodelos UML



# Perfiles UML

**UML no es lo suficientemente expresivo para representar conceptos específicos de dominios particulares. Por esta razón, UML incluye un mecanismo para extender y adaptar UML a diferentes dominios y plataformas: el "Perfil UML"**



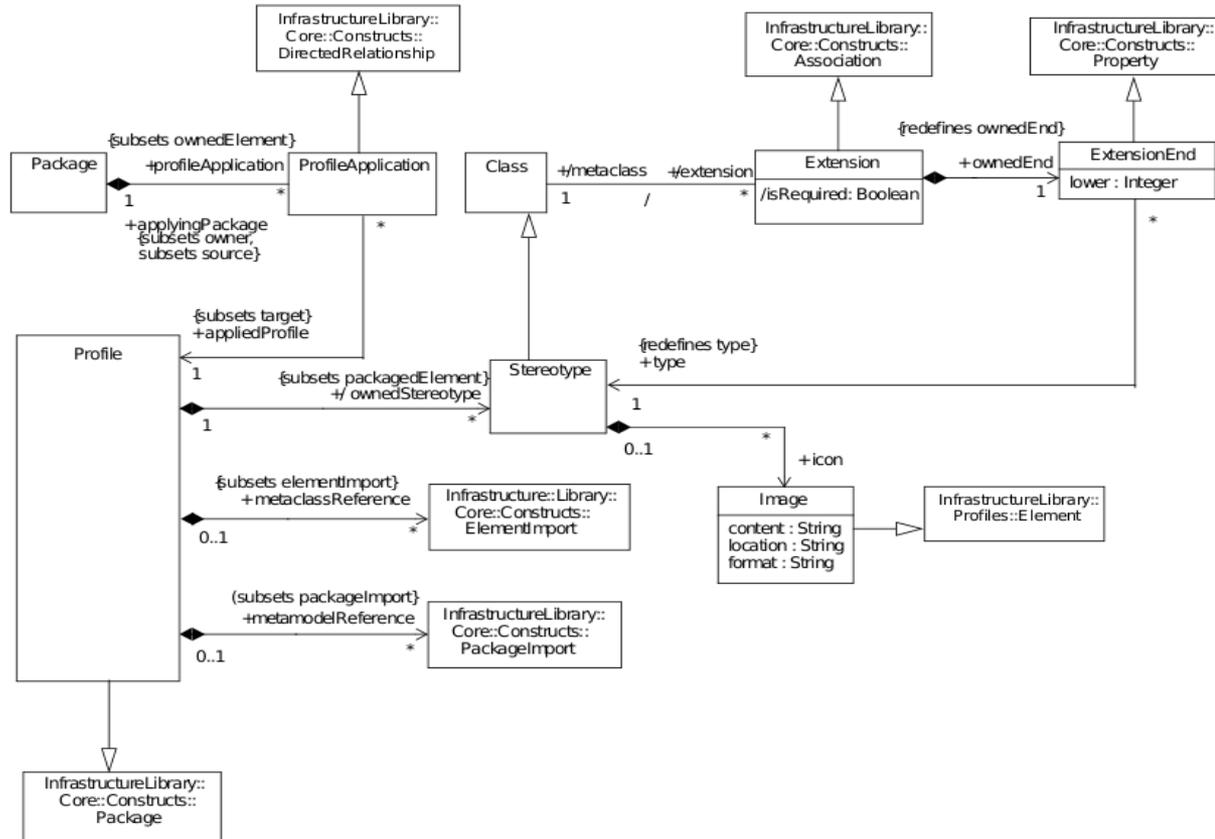
- Incluye tres mecanismos de extensión
  - Estereotipos → Extienden el vocabulario UML
  - Valores etiquetados → Especifican propiedades
  - Restricciones → Lenguaje Natural  
→ **OCL (Object Constraint Language)**

# Perfiles UML

## Ventajas

- Definir nuevo vocabulario para un dominio o plataforma específica.
- Dar sintaxis para constructores que no tienen notación.
- Definir notación particular para elementos ya existentes.
- Agregar semántica a elementos con semántica imprecisa.
- Agregar nueva semántica.
- Agregar información en modelos para su transformación.

# Perfiles UML



# Perfiles UML

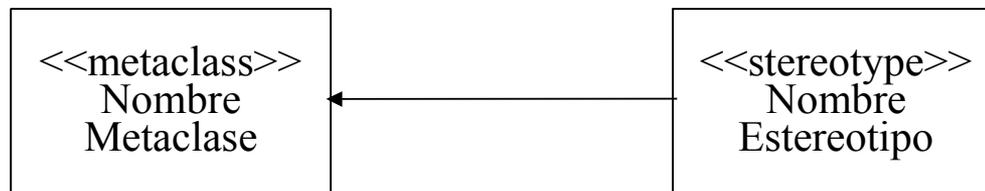
## Elementos

- Incluye tres mecanismos de extensión
  - Estereotipos
  - Valores etiquetados
  - Restricciones

# Perfiles UML

## Estereotipos

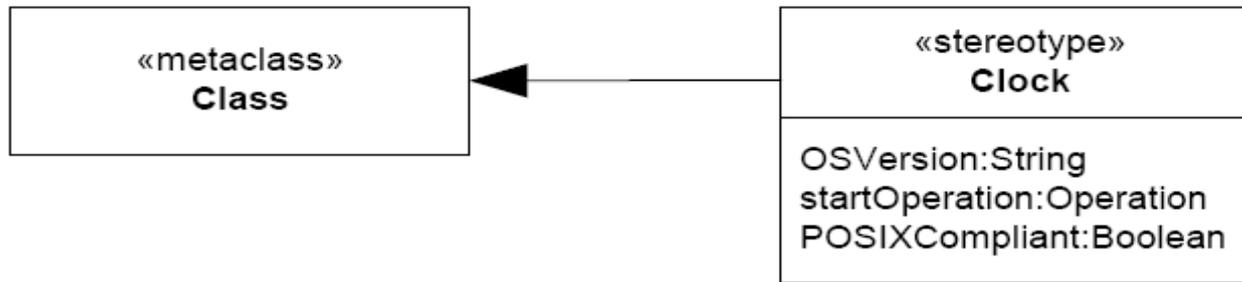
- Permiten añadir semántica a un elemento “tipos especiales de” elementos del mismo tipo
- Basados en cualquier tipo de elemento (clases, nodos, componentes, paquetes, relaciones, asociaciones, generalizaciones, dependencias)
- Se pueden utilizar en las mismas situaciones que el elemento original



# Perfiles UML

## Estereotipos

Definición



Uso



# Perfiles UML

## Estereotipos Estándar

### Nombre

<<auxiliary>>  
<<call>>  
<<create>>  
<<derive>>  
<<destroy>>  
<<document>>  
<<entity>>  
<<executable>>  
<<file>>  
<<focus>>  
<<framework>>  
<<instantiate>>  
<<library>>  
<<metaclass>>  
<<modelLibrary>>

### Aplicado a

Class  
Usage  
BehavioralFeature  
Abstraction  
BehavioralFeature  
Artifact  
Component  
Artifact  
Artifact  
Class  
Package  
Usage  
Artifact  
Class  
Package

### Nombre

<<realization>>  
<<refine>>  
<<responsability>>  
<<script>>  
<<send>>  
<<service>>  
<<source>>  
<<specification>>  
<<subsystem>>  
<<trace>>  
<<type>>  
<<utility>>  
<<process>>  
<<implement>>  
<<implementationClass>>

### Aplicado a

Classifier  
Abstraction  
Usage  
Artifact  
Usage  
Component  
Artifact  
Classifier  
Component  
Abstraction  
Class  
Class  
Component  
Component  
Class

# Perfiles UML

## Estereotipos

### Nombre

<<auxiliary>>  
<<call>>  
<<create>>  
<<derive>>  
<<destroy>>  
<<document>>  
<<entity>>  
<<executable>>  
<<file>>  
<<focus>>  
<<framework>>  
<<instantiate>>  
<<library>>  
<<metaclass>>  
<<modelLibrary>>

### Aplicado a

Class  
Usage  
BehavioralFeature  
Abstraction  
BehavioralFeature  
Artifact  
Component  
Artifact  
Artifact  
Class  
Package  
Usage  
Artifact  
Class  
Package

### Nombre

<<realization>>  
<<refine>>  
<<responsability>>  
<<script>>  
<<send>>  
<<service>>  
<<source>>  
<<specification>>  
<<subsystem>>  
<<trace>>  
<<type>>  
<<utility>>  
<<process>>  
<<implement>>  
<<implementationClass>>

### Aplicado a

Classifier  
Abstraction  
Usage  
Artifact  
Usage  
Component  
Artifact  
Classifier  
Component  
Abstraction  
Class  
Class  
Component  
Component  
Class

# Perfiles UML

## Estereotipos

`<<interface>>` especifica una colección de operaciones

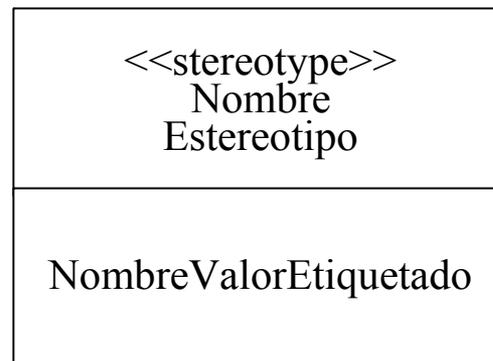
`<<type>>` especifica estructura y comportamiento sin considerar implementación

`<<enumeration>>` especifica valores discretos

# Perfiles UML

## Valores etiquetados

- Al igual que las clases, los estereotipos pueden tener propiedades. Cuando un estereotipo es aplicado a un elemento de modelo, los valores que pueden tomar los estereotipos son marcados como valores etiquetados.



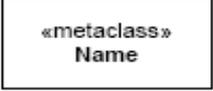
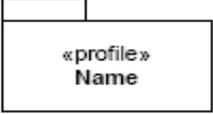
# Perfiles UML

## Restricciones

- Restricciones semánticas asociadas al perfil
  - Lenguaje natural
  - OCL

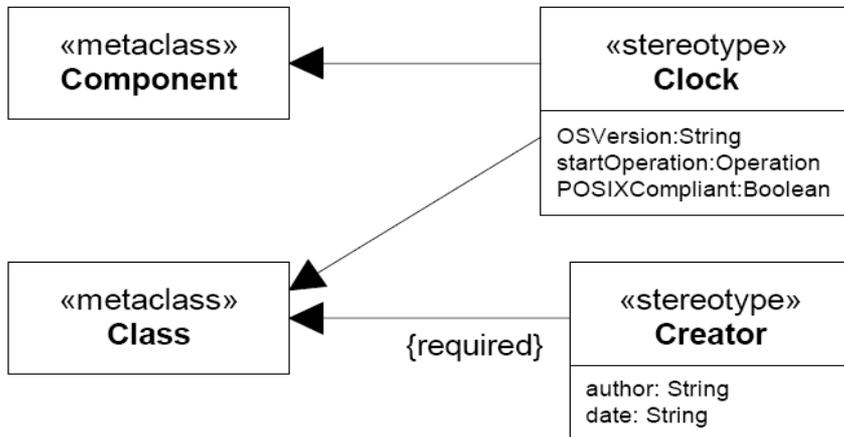
# Definición de Perfiles UML

## Notación gráfica (elementos estructurales)

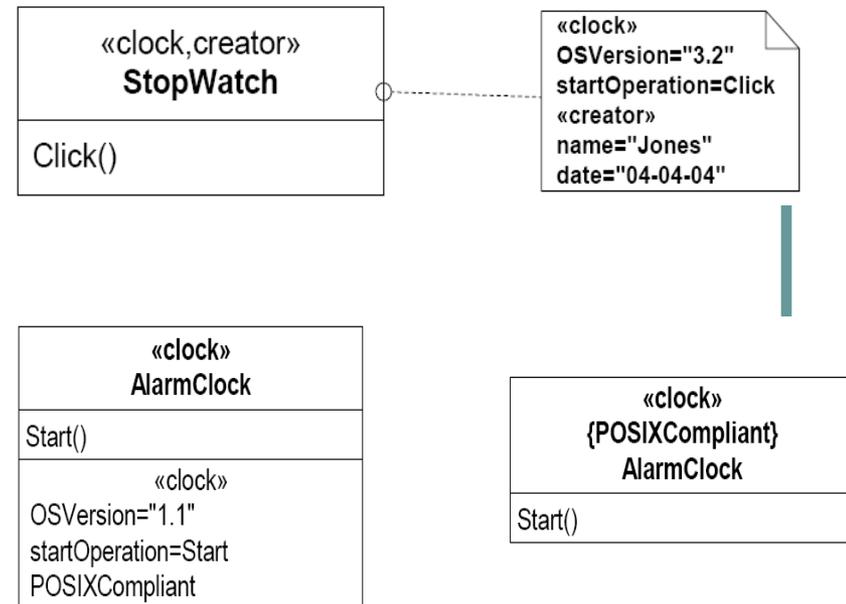
Stereotype	
Metaclass	
Profile	

# Definición de Perfiles UML

## Definición



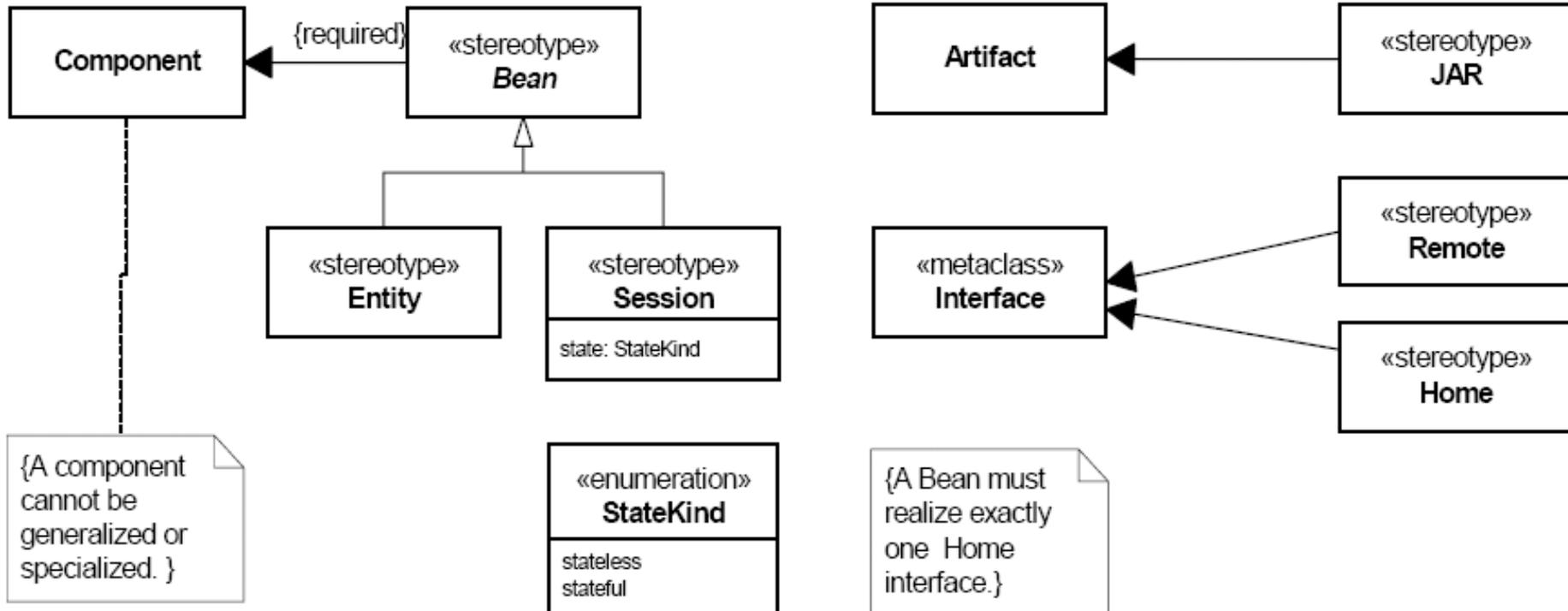
## Uso



# Definición de Perfiles UML

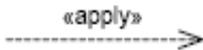
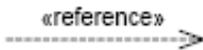
## Ejemplo

«profile» EJB

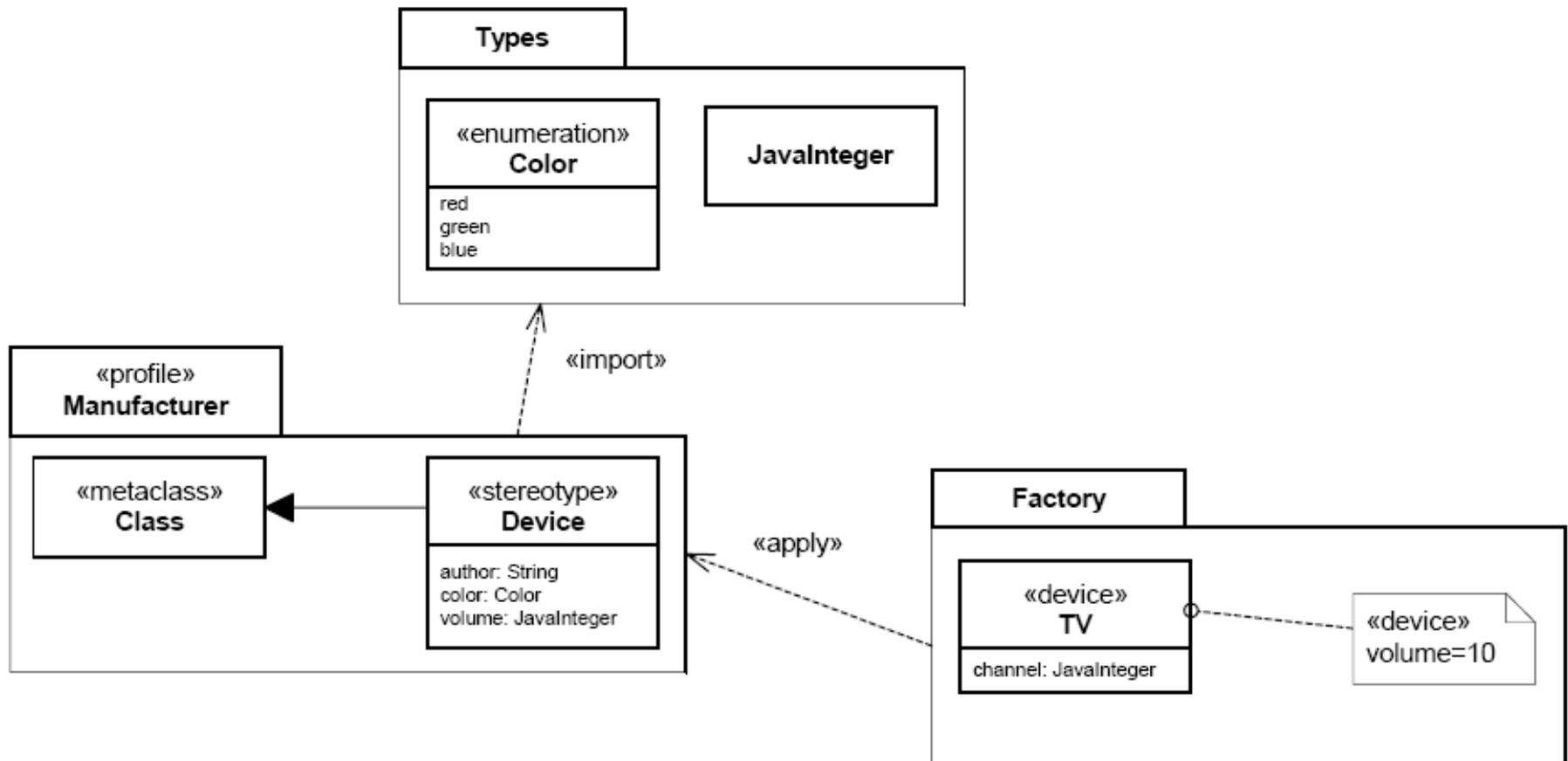


# Definición de Perfiles UML

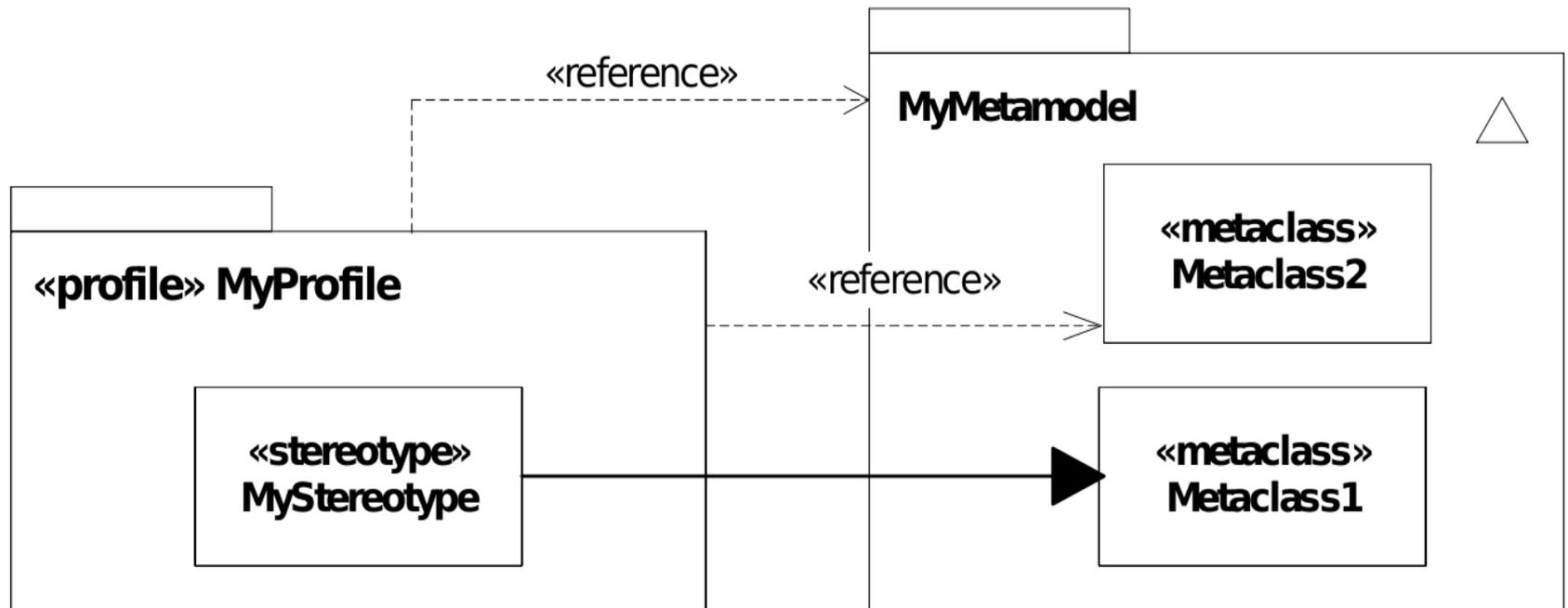
## Dependencias

ProfileApplication	
ElementImport PackageImport	

# Definición de Perfiles



# Definición de Perfiles UML



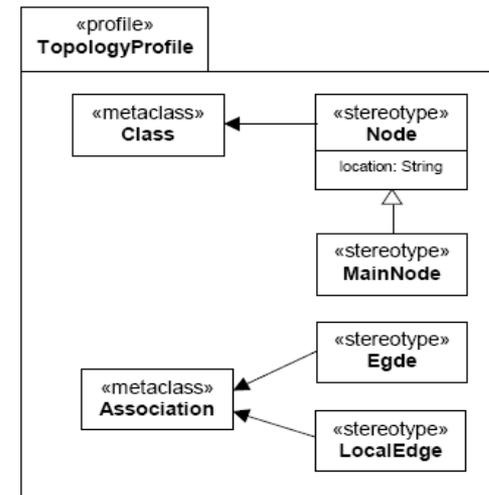
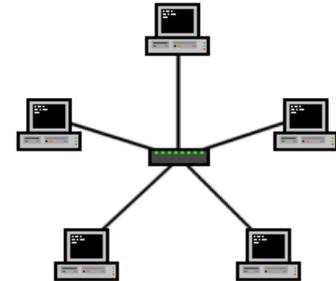
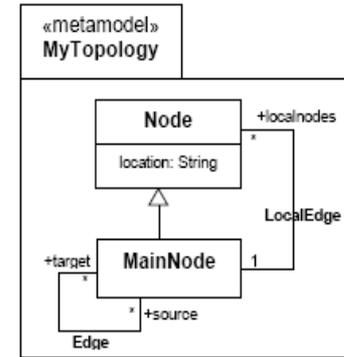
# Construcción de Perfiles UML

(1) Definir el metamodelo del dominio de aplicación a modelar con un Perfil. Si no existiese, entonces definiríamos dicho metamodelo utilizando los mecanismos del propio UML (clases, relaciones de herencia, asociaciones, etc.).

(2) Definir el perfil usando un paquete «profile». Incluir un estereotipo por cada uno de los elementos del metamodelo que deseamos incluir en el Perfil. Estos estereotipos tendrán el mismo nombre que los elementos del metamodelo.

(3) Definir cuáles son los elementos del metamodelo de UML que estamos extendiendo sobre los que es posible aplicar un estereotipo (clases, sus asociaciones, sus atributos, las operaciones, las transiciones, los paquetes, etc.)

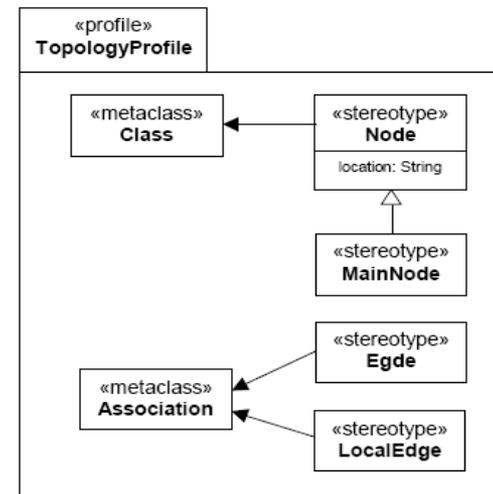
(4) Asociar metaclases con estereotipos.



# Construcción de Perfiles UML

(5) Definir como valores etiquetados de los elementos del Perfil los atributos que aparezcan en el metamodelo. Incluir la definición de sus tipos, y sus posibles valores iniciales.

(6) Definir las restricciones que forman parte del Perfil, a partir de las restricciones del dominio. Por ejemplo, las multiplicidades de las asociaciones que aparecen en el metamodelo del dominio, o las propias reglas de negocio de la aplicación deben traducirse en la definición las correspondientes restricciones.



# Definición de Perfiles

## Perfiles UML publicados por OMG

- UML Profile for CORBA
- UML Profile for CORBA® Component Model (CCM)
- UML Profile for Data Distribution
- UML Profile for Enterprise Application Integration (EAI)
- UML Profile for Enterprise Distributed Object Computing (EDOC)
- UML Profile for Modeling and Analysis of Real-time and Embedded Systems (MARTE)
- UML Profile for Modeling QoS and Fault Tolerance Characteristics and Mechanisms
- UML Profile for Schedulability, Performance and Time
- UML Profile for Software Radio
- UML Profile for System on a Chip (SoC)
- UML Profile for Voice
- UML Testing Profile

# Definición de Perfiles

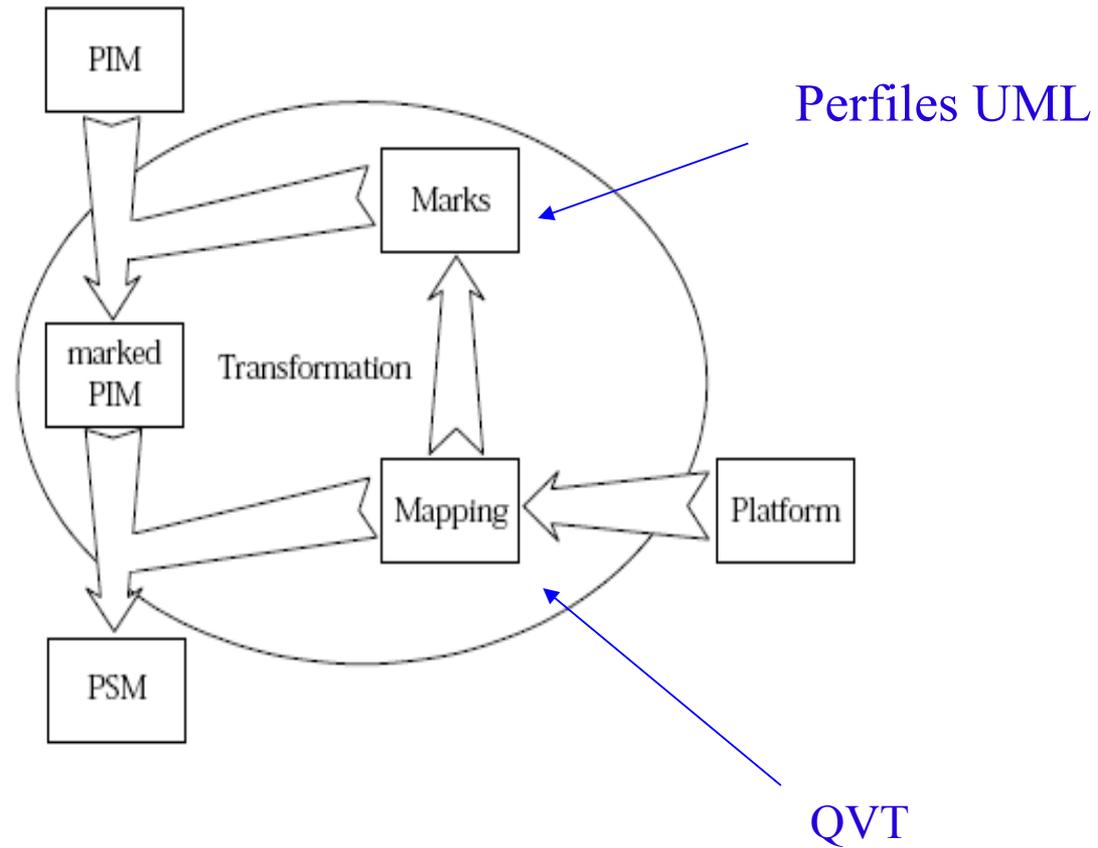
## • Otros Perfiles UML

- Enterprise Java Beans (by Java Community Process)
- Software Services (by IBM, supported by Rational Software Architect UML CASE tool)
- Knowledge-Based Systems (University of York)
- Data Modeling (by [agiledata.org](http://agiledata.org))
- Framework Architectures (UML-F)
- Requirement Engineering with KAOS
- Formal Methods in B (UML-B)
- Embedded System Design

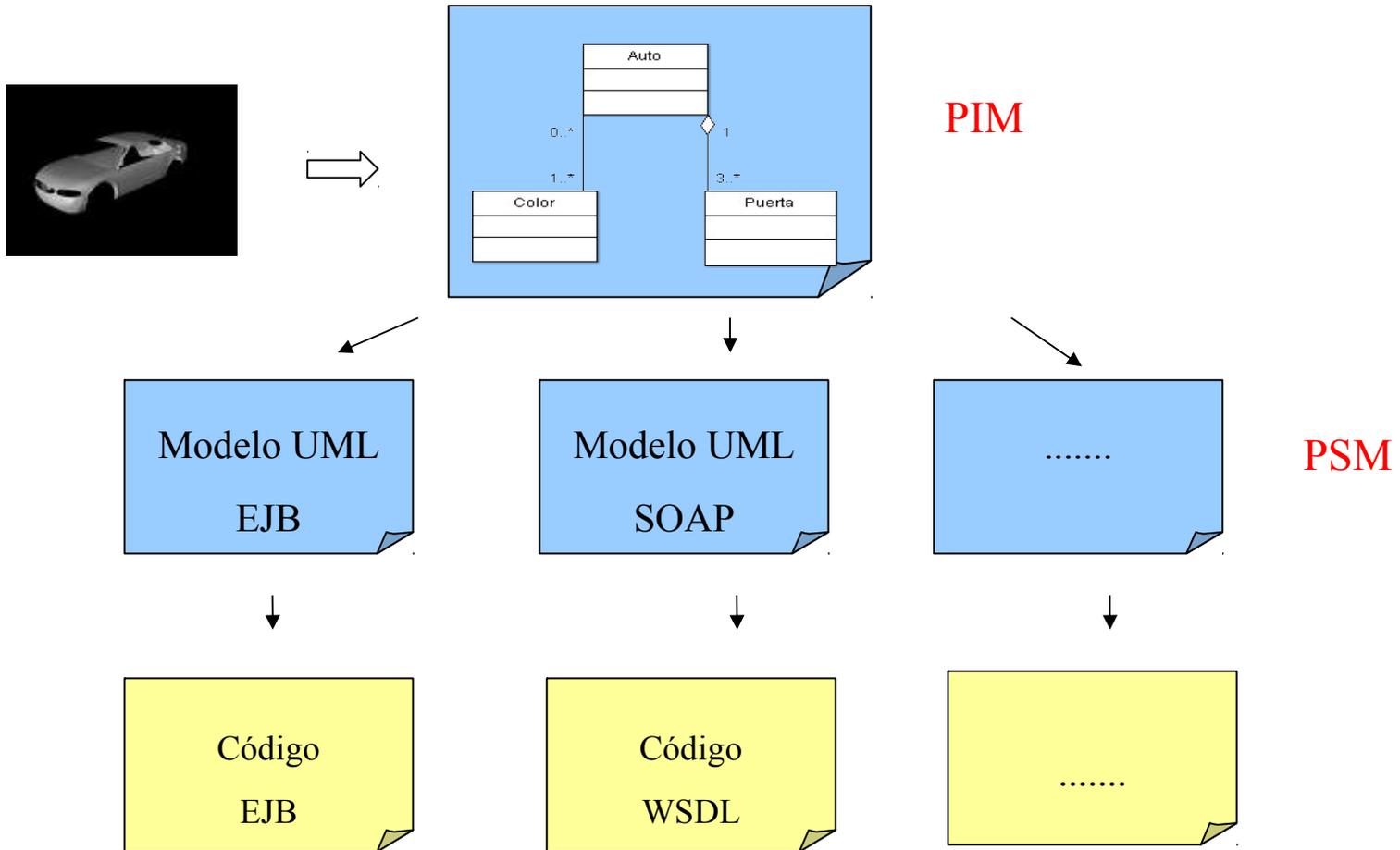
## **Unified Modeling Language (UML)**

- Introducción a UML
- Diagramas UML
- Diagramas de clases
- Metamodelado
- Perfiles UML
- **Perfiles en MDA**

# Perfiles UML en MDA

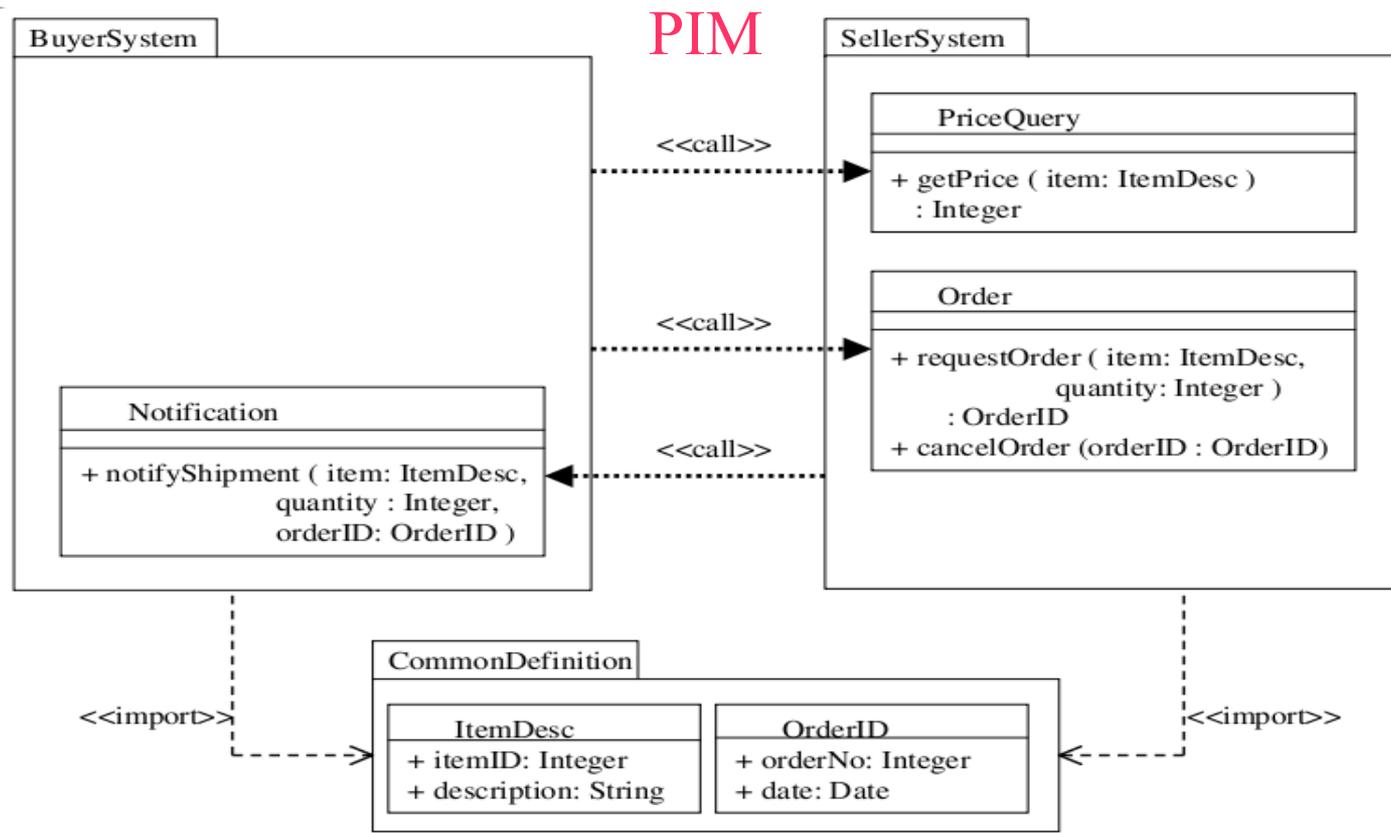


# Perfiles UML en MDA

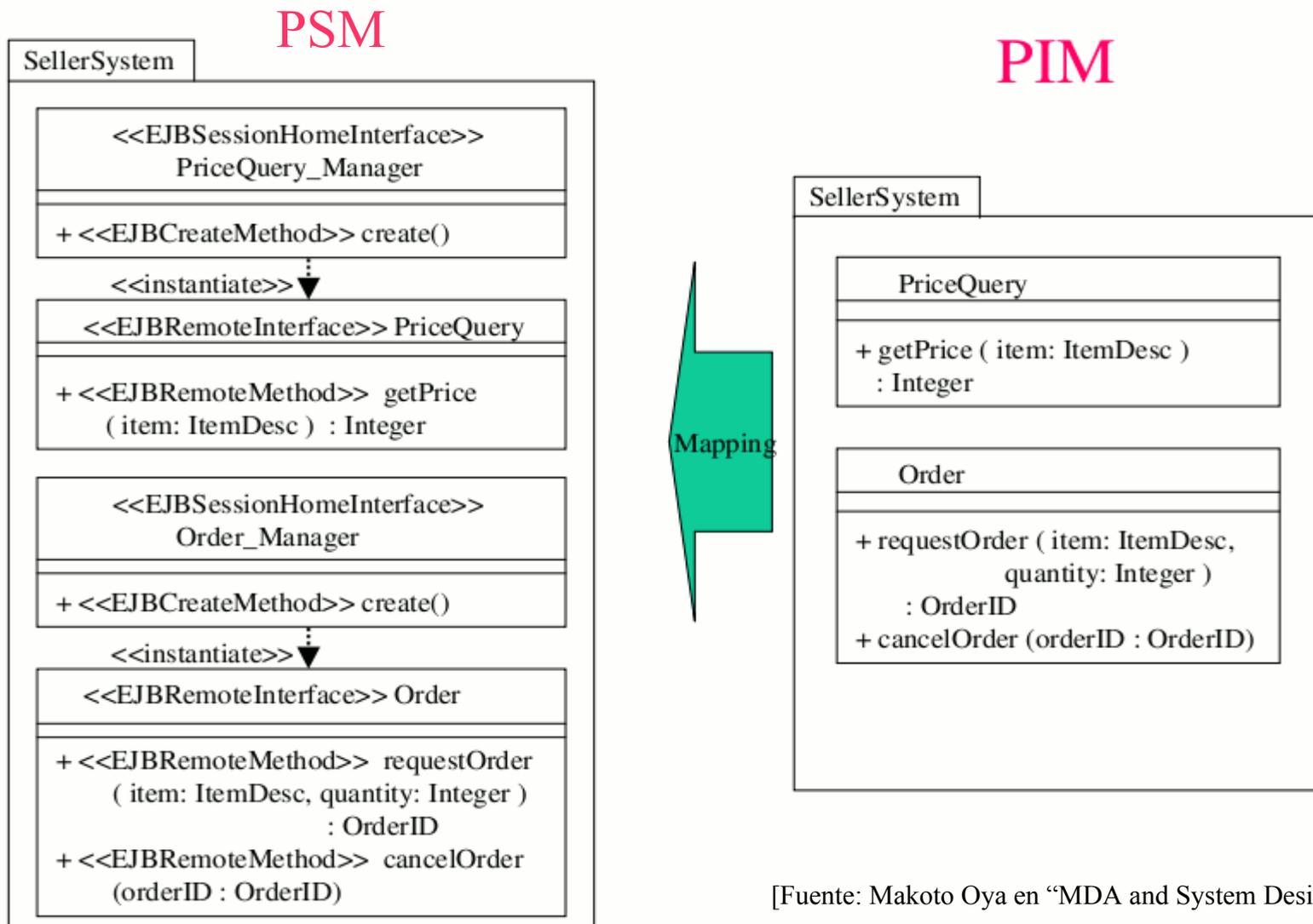


# Aplicación de Perfiles UML

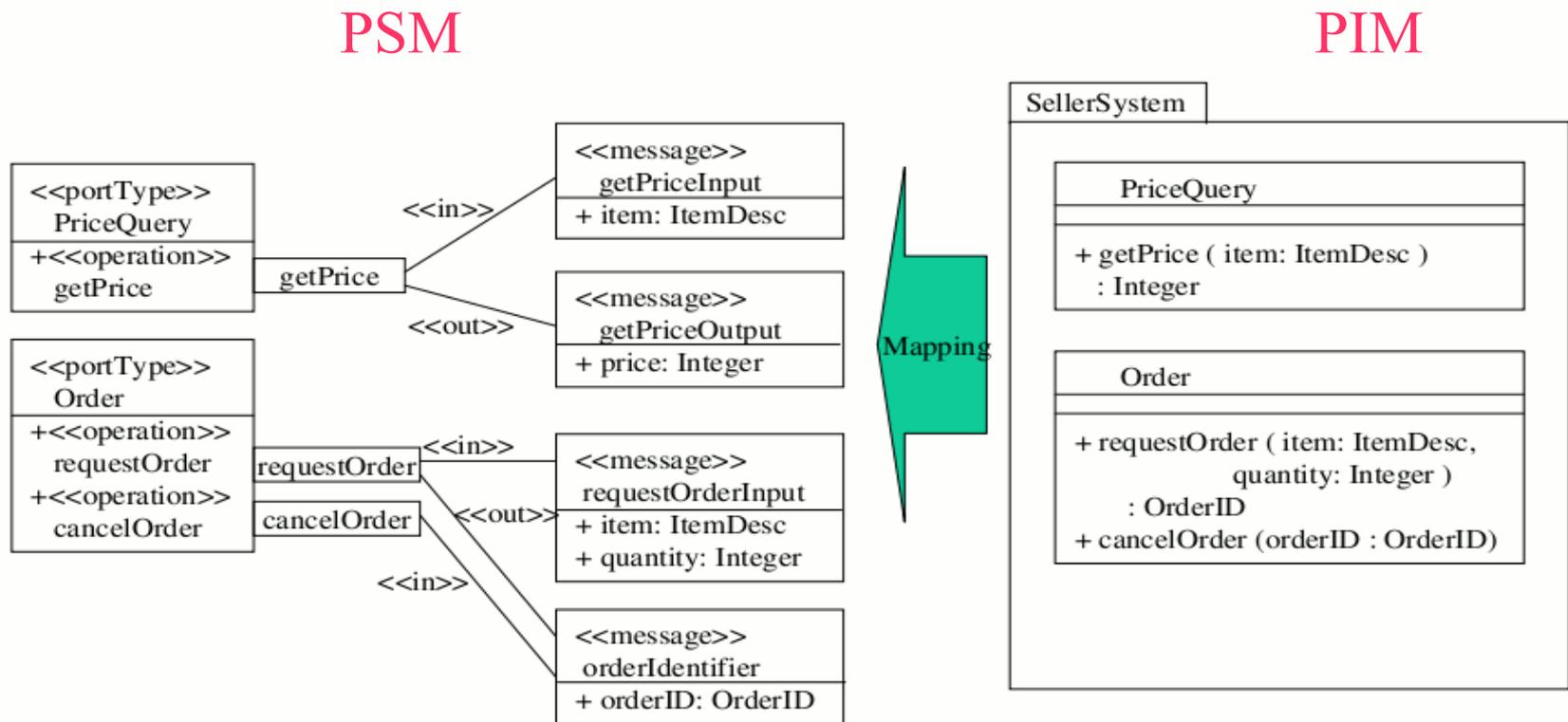
## Ejemplo



# Aplicación de Perfiles UML



# Aplicación de Perfiles UML



# Perfiles UML y Herramientas MDA

- Existen herramientas MDA que contemplan la definición de Perfiles
- Sin embargo, no existe portabilidad entre perfiles definidos con distintas herramientas